REMARKS

Claims 2-8 are pending in the present application. The Office Action and cited references have been considered. Favorable reconsideration is respectfully requested.

The Examiner is thanked for the courtesies shown during the interview of October 26, 2009. This amendment is presented in accordance with the discussions during that interview.

Claims 2-4 and 8 were rejected under 35 U.S.C. §102(b) as being anticipated by Cirne (U.S. Patent No. 6,260,187 - hereinafter "Cirne"). Claims 5 and 6 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cirne in view of official notice. These rejections are respectfully traversed for the following reasons.

Claim rejections — 35 USC §102(b)

Claim 7 recites a method for loading into a computer device, an updated release of an earlier application written in a programming language using objects, having earlier application classes and earlier static field identifiers, the updated release having updated application release classes, and updated static field identifiers, said earlier application having stored persistent data in the form of objects of said earlier classes or in static fields identified by said earlier static field identifiers, the method comprising the steps of computing, in a first computing operation prior to the loading, a class matching information establishing a correspondence between the earlier application release classes and the updated application release classes; computing, in a second computing operation prior to the loading, a second static field identifiers matching information establishing a correspondence between the earlier application release static field identifiers and the

updated application release static field identifiers; linking the class matching information and the static field identifiers matching information to the updated application release as loaded into the computer device; and using the class matching information and the static field identifiers matching information to modify the objects to point at the updated application release classes and use the updated application release static field identifiers. This is not taught, disclosed or made obvious by the prior art of record.

Claim 7 is directed to a method for loading an updated release of an earlier application into a computer device, said earlier application having earlier application classes and earlier static field identifiers and said updated release having updated application classes and updated static field identifiers.

The wording of claim 7 has been modified to clarify the fact that the application was previously loaded in the computer device, not only in the sense that its code was present in some storage medium or memory of the device, but in the stronger sense that the application had previously been executed on the computer device and that data (in the form of objects) of the earlier application release is present in some storage medium or memory of the computer device. For example, in the case of a chip card application as described by Applicant, it is essential that an update to the application conserves or adapts the data stored by the earlier application release, rather than deleting and reloading the data (specification of the published application no. 2007/0277168, [0005]). Loading the updated application on the computer device comprises not only copying the code into a storage medium or memory of this computer device, but also initializing static fields of the updated classes (using already stored values for inherited fields)

(specification, [0063]) and modifying existing objects to make use of the updated classes (specification, [0064]).

The method recited in claim 7 comprises four steps, here numbered for easy reference:

1. computing in a first computing operation prior to said loading a class matching information establishing a correspondence between said earlier application release classes and said updated application release classes;
2. computing in a second computing operation, prior to said loading a second static field identifiers matching information establishing a correspondence between said earlier application release static field identifiers and said updated application release static field identifiers;
3. linking said class matching information and said static field identifiers matching information to said updated application release as loaded into the computer device; and
4. using said class matching information and said static field identifiers matching information to modify said objects to point at the updated application release classes and use the updated application release static field identifiers.

Steps 1 and 2 recite the computing of correspondences between characteristics of the earlier application release and the updated application release.

Step 3 recites the linking of the correspondences computed in steps 1–2 "to said updated application release as loaded into the computer device".

Step 4 recites the use of the correspondences computed in steps 1–2 to modify objects of the earlier application release so that they become objects of the updated application release. That is to say, objects already residing on the computer device are modified to use the updated application classes (specification, [0064]).

During the interview, the Examiner suggested that Applicant amend claim 7 to recite "during the run-time execution of the updated application". Applicant has amended claim 7 to include the qualifier "persistent" with regard to the stored data. Applicant respectfully submits that this amendment follows the Examiner's suggested claim limitation, while keeping with the terminology of the specification as originally filed (specification, [0003]).

The goal of the invention is, in general terms, to permit updating an application update on a chip card or other computing device. More specifically, as described in the specification, the invention has three desirable properties:

The invention makes it possible to modify the code of the program, for example by replacing the code of a procedure or method by different code (*e.g.*, to correct a defect), or by adding new code (*e.g.*, to provide new functionality) (specification, [0007]).

The invention makes it possible to modify the structure of the program; in a programming language using objects, this means modifying the class hierarchy (specification, [0008]).

The invention makes it possible to modify the structure of persistent data in the form of existing objects that were stored on the computing device prior to the application update (specification [0003]), in order to adapt these existing objects to the new class hierarchy of the updated application (specification, [0009]). This third property is often a requirement on chip

cards, as security considerations preclude copying data off the chip card and back (specification, [0005]).

Cirne teaches a method for loading into a computer device an updated release of an earlier application (Cirne, abstract; Cirne, col. 2, l. 45–47). More precisely, Cirne teaches a method for transforming the code of an application according to provided rules to produce an updated release of that application. Cirne teaches modifying the code and structure of an application written in a programming language using object through the use of rules such as substitute class rules, proxy class rules and substitute static field rules (col. 3, l. 65 to col. 4, l. 9; col. 4, l. 20–25; col. 4, l. 36–44). In this manner, Cirne's method enjoys the first and second desirable properties described in the above introductory remarks.

Cirne does not teach modifying the structure of existing objects produced by the application being updated prior to that application being updated. In other words, Cirne does not teach how to handle persistent data when an application is updated. Indeed, Cirne discusses transforming the code of the application being updated, but not executing that code. Therefore Cirne's disclosure does not address the third desirable property described above.

Claim 7 recites updating an application "having stored persistent data in the form of objects of said earlier classes or in static fields identified by said earlier static field identifiers". As explained above, Cirne discloses earlier classes and earlier static field identifiers; but as Cirne does not discuss the execution of the earlier release of the application, Cirne does not disclose having stored persistent data in a form consistent with said earlier release such as objects of earlier classes or static fields identified by earlier static field identifiers. Thus this limitation is not present in the prior art cited by the Examiner.

Claim 7 further recites as part of its fourth and last step "modify said objects to point at the updated application release classes and use the updated application release static field identifiers". Cirne discloses modifying code according to substitute class rules and substitute static field rules in order to obtain updated code that uses an updated class structure and updated static fields (col. 2, l. 50–67; col. 5, l. 29–58; fig. 6, 8). Cirne also discloses modifying code that creates a new object so that in the updated application uses and updated class structure and updated static fields (col. 11, l. 28–49). Cirne's disclosure concerns a modification of code in order to comply with an updated application structure. Cirne does not disclose modifying existing objects in any way. Hence the fourth step of claim 7 is not anticipated by Cirne.

As per claim 2-4 and 8, Claims 2-4 and 8 must be read in view of claim 1. Then, Kuhn does not disclose claims 2-4 and 8.

Claim rejections — 35 USC §103(a)

As Cirne does not even suggest that data may be modified as part of an application update, then, step 4 is not even suggested by Cirne, and the subject matter of claim 7 would not have been obvious to one having ordinary skill in the art. Claims 5 and 6 must be read in view of claim 1. Then, Cirne does not disclose nor suggest subject matter of claim 5 or 6. Then subject matter of claim 5 or 6 would not have been obvious to one having ordinary skill in the art.

Cirne teaches a method for transforming program code before it is run. In particular, Cirne does not teach running or even loading both the original program code and the updated program code on the same computer device. On the other hand, Applicant's method

concerns loading and executing an updated application on a computer device on which the original application was previously loaded and, preferably, run (¶0011).

Cirne does not teach nor suggest a method for loading an updated version of an original one, while this original application runs.

Crucially, Cirne only teaches a transformation of program code, whereas Applicant's invention deals with a transformation of program data. Applicant's invention comprises a step of modifying object data stored on the computer device on which the original program was loaded in order for the resulting data to be suitable for the updated application (¶0011; claim 7). This point is very relevant in practice because chip card applications may store confidential data (such as secret cryptographic keys) that never leaves the chip card after it has been produced under highly secure conditions (¶0005).

Cirne does not teach nor suggest a method for loading an updated version of an original one, without uploading and/or downloading confidential data already stored on the chip card.

Applicant respectfully submits that the purpose of the method according to Cirne is quite different from that of Applicant's method. For this reason, the teaching of Cirne would not have been used by one of ordinary skill in the art to solve the problems faced by Applicant.

In view of the above amendment and remarks, Applicant respectfully requests reconsideration and withdrawal of the outstanding rejections of record. Applicant submits that the application is in condition for allowance and early notice to this effect is most earnestly solicited.

If the Examiner has any questions, he is invited to contact the undersigned at 202-628-5197.

Respectfully submitted,

BROWDY AND NEIMARK, P.L.L.C.
Attorneys for Applicant(s)

By /Ronni S. Jillions/
Ronni S. Jillions
Registration No. 31,979

RSJ:me
Telephone No.: (202) 628-5197
Facsimile No.: (202) 737-3528
G:\BN\M\Mout\Vetillard1\Pto\2009-12-21mendmentVETILLARD1.doc